


```

RRRRRRRR RRRRRRRR P PPPPPP P PPPPPP GGGGGGGG DDDDDDDD S SSSSSSSS P PPPPPP LL YY
RRRRRRRR RRRRRRRR R R R R P P PP PP GG DD DD SS SS PP PP LL YY
RRRRRRRR RRRRRRRR R R R R P P PP PP GG DD DD SS SS PP PP LL YY
RRRRRRRR RRRRRRRR R R R R P P PP PP GG DD DD SS SS PP PP LL YY
RRRRRRRR RRRRRRRR P PPPPPP P PPPPPP GG DD DD SSSSSS P PPPPPP LL YY
RRRRRRRR RRRRRRRR P PPPPPP P PPPPPP GG DD DD SSSSSS P PPPPPP LL YY
RRRRRRRR RRRRRRRR R R R R P P PP GG GGGGGG DD DD SS SS PP LL YY
RRRRRRRR RRRRRRRR R R R R P P PP GG GGGGGG DD DD SS SS PP LL YY
RRRRRRRR RRRRRRRR R R R R P P PP GG GG DD DD SS SS PP LL YY
RRRRRRRR RRRRRRRR R R R R P P PP GGGGGG DDDDDDDD SSSSSSSS PP LL LLLLLLLL YY
RRRRRRRR RRRRRRRR R R R R P P PP GGGGGG DDDDDDDD SSSSSSSS PP LL LLLLLLLL YY

```

```
1 0001 0 MODULE RPG$DSPLY( %TITLE 'DSPLY an item'  
2 0002 0 IDENT = '1-004'  
3 0003 0 ) =  
4 0004 1 BEGIN  
5 0005 1  
6 0006 1 *****  
7 0007 1 *  
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY  
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.  
10 0010 1 * ALL RIGHTS RESERVED.  
11 0011 1 *  
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
17 0017 1 * TRANSFERRED.  
18 0018 1 *  
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
21 0021 1 * CORPORATION.  
22 0022 1 *  
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.  
25 0025 1 *  
26 0026 1 *  
27 0027 1 *****  
28 0028 1  
29 0029 1  
30 0030 1 ++  
31 0031 1  
32 0032 1 FACILITY: RPGII SUPPORT  
33 0033 1  
34 0034 1 ABSTRACT:  
35 0035 1  
36 0036 1 This module contains the RTL routine that supports the RPG DSPLY  
37 0037 1 opcode.  
38 0038 1  
39 0039 1 ENVIRONMENT: VAX/VMS user mode  
40 0040 1  
41 0041 1 AUTHOR: Debess Grabazs, CREATION DATE: 18-March-1983  
42 0042 1  
43 0043 1 MODIFIED BY:  
44 0044 1  
45 0045 1 1-001 - Original version. DG 18-Mar-1983.  
46 0046 1 1-002 - use routine COB$$_RET_A_AB_PREV to get address of COB$$AB_PREV.  
47 0047 1 MDL 29-Aug-1983  
48 0048 1 1-003 - Create temporary for overpunched numeric before calling  
49 0049 1 COB$ACC_SCR. In case of invalid entry, original value will  
50 0050 1 be saved. DG 30-Aug-1983.  
51 0051 1 1-004 - Set RPG for calls to both COB$ACC_SCR and COB$DISP_SCR.  
52 0052 1 DG 24-Oct-1983.  
53 0053 1  
54 0054 1 --  
55 0055 1 <BLF/PAGE>
```

```
57 0056 1 %SBTTL 'Declarations'
58 0057 1 +
59 0058 1 PROLOGUE FILE:
60 0059 1 !-
61 0060 1
62 0061 1 REQUIRE 'RTLIN:RPGPROLOG';                                ! Switches, PSECTs, macros,
63 0126 1                                                               linkages and LIBRARYs
64 0127 1
65 0128 1 +
66 0129 1 LINKAGES
67 0130 1 NONE
68 0131 1 !-
69 0132 1
70 0133 1 +
71 0134 1 TABLE OF CONTENTS:
72 0135 1 !-
73 0136 1
74 0137 1 FORWARD ROUTINE
75 0138 1 RPG$DSPLY: NOVALUE;
76 0139 1
77 0140 1 +
78 0141 1 INCLUDE FILES
79 0142 1 NONE
80 0143 1 !-
81 0144 1
82 0145 1 +
83 0146 1 MACROS
84 0147 1 NONE
85 0148 1 !-
86 0149 1
87 0150 1 +
88 0151 1 EQUATED SYMBOLS
89 0152 1 NONE
90 0153 1 !-
91 0154 1
92 0155 1 +
93 0156 1 EXTERNAL REFERENCES
94 0157 1 !-
95 0158 1
96 0159 1 EXTERNAL ROUTINE
97 0160 1 COB$ACC_SCR,
98 0161 1 COB$DISP_SCR,
99 0162 1 COB$SRET_A_AB_PREV,                                ! ACCEPT with conversion
100 0163 1
101 0164 1 LIB$SIGNAL;                                ! DISPLAY with conversion
102 0165 1
103 0166 1 EXTERNAL LITERAL
104 0167 1 RPG$_INVNUMENT;                                ! Return address of table giving
105 0168 1
106 0169 1 EXTERNAL
107 0170 1 RPGSBTZ;                                ! call history for advancing purposes
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
988
989
989
990
991
992
993
994
995
996
997
997
998
999
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1097
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1189
1190
1191
1192
1193
1194
1195
1196
1197
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1297
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1397
1397
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1497
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1597
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1697
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1797
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1897
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1997
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096
2097
2097
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2189
2190
2191
2192
2193
2194
2195
2196
2197
2197
2198
2199
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2209
2210
2
```

```
109      0171 1 %SBTTL 'RPGSDSPLY - DSPLY an item'  
110      0172 1 GLOBAL ROUTINE RPGSDSPLY(  
111      0173 1  
112      0174 1      FLAGS,  
113      0175 1      DISPLAY FIELD: REF BLOCK[,BYTE],  
114      0176 1      ACCEPT FIELD: REF BLOCK[,BYTE]  
115      0177 1      ): NOVALUE =  
116      0178 1      ++  
117      0179 1  
118      0180 1      FUNCTIONAL DESCRIPTION:  
119      0181 1  
120      0182 1      This routine supports the RPG DSPLY opcode.  
121      0183 1  
122      0184 1      CALLING SEQUENCE:  
123      0185 1  
124      0186 1      CALL RPGSDSPLY (flags.fl.v [,display_field.rx.dx]  
125      0187 1      [,accept_field.mx.dx])  
126      0188 1  
127      0189 1      FORMAL PARAMETERS:  
128      0190 1  
129      0191 1      flags  
130      0192 1      bit 0 is set on if a comma should be  
131      0193 1      used in place of decimal point; bit 1  
132      0194 1      is set on if blanks in an overpunched  
133      0195 1      numeric field should be treated as  
134      0196 1      equivalent to zeroes.  
135      0197 1      display_field  
136      0198 1      data item whose value is displayed.  
137      0199 1  
138      0200 1      accept_field  
139      0201 1      data item whose value is displayed and  
140      0202 1      a new value is accepted into.  
141      0203 1  
142      0204 1  
143      0205 1  
144      0206 1  
145      0207 1  
146      0208 1  
147      0209 1  
148      0210 1  
149      0211 1  
150      0212 1  
151      0213 1  
152      0214 1  
153      0215 1  
154      0216 1  
155      0217 1  
156      0218 1      SIDE EFFECTS:  
157      0219 1      NONE  
158      0220 1  
159      0221 1  
160      0222 1  
161      0223 1  
162      0224 1  
163      0225 1  
164      0226 1  
165      0227 1  
166      0228 1  
167      0229 1  
168      0230 1  
169      0231 1  
170      0232 1  
171      0233 1  
172      0234 1  
173      0235 1  
174      0236 1  
175      0237 1  
176      0238 1  
177      0239 1  
178      0240 1  
179      0241 1  
180      0242 1  
181      0243 1  
182      0244 1  
183      0245 1  
184      0246 1  
185      0247 1  
186      0248 1  
187      0249 1  
188      0250 1  
189      0251 1  
190      0252 1  
191      0253 1  
192      0254 1  
193      0255 1  
194      0256 1  
195      0257 1  
196      0258 1  
197      0259 1  
198      0260 1  
199      0261 1  
200      0262 1  
201      0263 1  
202      0264 1  
203      0265 1  
204      0266 1  
205      0267 1  
206      0268 1  
207      0269 1  
208      0270 1  
209      0271 1  
210      0272 1  
211      0273 1  
212      0274 1  
213      0275 1  
214      0276 1  
215      0277 1  
216      0278 1  
217      0279 1  
218      0280 1  
219      0281 1  
220      0282 1  
221      0283 1  
222      0284 1  
223      0285 1  
224      0286 1  
225      0287 1  
226      0288 1  
227      0289 1  
228      0290 1  
229      0291 1  
230      0292 1  
231      0293 1  
232      0294 1  
233      0295 1  
234      0296 1  
235      0297 1  
236      0298 1  
237      0299 1  
238      0200 1  
239      0201 1  
240      0202 1  
241      0203 1  
242      0204 1  
243      0205 1  
244      0206 1  
245      0207 1  
246      0208 1  
247      0209 1  
248      0200 1  
249      0201 1  
250      0202 1  
251      0203 1  
252      0204 1  
253      0205 1  
254      0206 1  
255      0207 1  
256      0208 1  
257      0209 1  
258      0200 1  
259      0201 1  
260      0202 1  
261      0203 1  
262      0204 1  
263      0205 1  
264      0206 1  
265      0207 1  
266      0208 1  
267      0209 1  
268      0200 1  
269      0201 1  
270      0202 1  
271      0203 1  
272      0204 1  
273      0205 1  
274      0206 1  
275      0207 1  
276      0208 1  
277      0209 1  
278      0200 1  
279      0201 1  
280      0202 1  
281      0203 1  
282      0204 1  
283      0205 1  
284      0206 1  
285      0207 1  
286      0208 1  
287      0209 1  
288      0200 1  
289      0201 1  
290      0202 1  
291      0203 1  
292      0204 1  
293      0205 1  
294      0206 1  
295      0207 1  
296      0208 1  
297      0209 1  
298      0200 1  
299      0201 1  
300      0202 1  
301      0203 1  
302      0204 1  
303      0205 1  
304      0206 1  
305      0207 1  
306      0208 1  
307      0209 1  
308      0200 1  
309      0201 1  
310      0202 1  
311      0203 1  
312      0204 1  
313      0205 1  
314      0206 1  
315      0207 1  
316      0208 1  
317      0209 1  
318      0200 1  
319      0201 1  
320      0202 1  
321      0203 1  
322      0204 1  
323      0205 1  
324      0206 1  
325      0207 1  
326      0208 1  
327      0209 1  
328      0200 1  
329      0201 1  
330      0202 1  
331      0203 1  
332      0204 1  
333      0205 1  
334      0206 1  
335      0207 1  
336      0208 1  
337      0209 1  
338      0200 1  
339      0201 1  
340      0202 1  
341      0203 1  
342      0204 1  
343      0205 1  
344      0206 1  
345      0207 1  
346      0208 1  
347      0209 1  
348      0200 1  
349      0201 1  
350      0202 1  
351      0203 1  
352      0204 1  
353      0205 1  
354      0206 1  
355      0207 1  
356      0208 1  
357      0209 1  
358      0200 1  
359      0201 1  
360      0202 1  
361      0203 1  
362      0204 1  
363      0205 1  
364      0206 1  
365      0207 1  
366      0208 1  
367      0209 1  
368      0200 1  
369      0201 1  
370      0202 1  
371      0203 1  
372      0204 1  
373      0205 1  
374      0206 1  
375      0207 1  
376      0208 1  
377      0209 1  
378      0200 1  
379      0201 1  
380      0202 1  
381      0203 1  
382      0204 1  
383      0205 1  
384      0206 1  
385      0207 1  
386      0208 1  
387      0209 1  
388      0200 1  
389      0201 1  
390      0202 1  
391      0203 1  
392      0204 1  
393      0205 1  
394      0206 1  
395      0207 1  
396      0208 1  
397      0209 1  
398      0200 1  
399      0201 1  
400      0202 1  
401      0203 1  
402      0204 1  
403      0205 1  
404      0206 1  
405      0207 1  
406      0208 1  
407      0209 1  
408      0200 1  
409      0201 1  
410      0202 1  
411      0203 1  
412      0204 1  
413      0205 1  
414      0206 1  
415      0207 1  
416      0208 1  
417      0209 1  
418      0200 1  
419      0201 1  
420      0202 1  
421      0203 1  
422      0204 1  
423      0205 1  
424      0206 1  
425      0207 1  
426      0208 1  
427      0209 1  
428      0200 1  
429      0201 1  
430      0202 1  
431      0203 1  
432      0204 1  
433      0205 1  
434      0206 1  
435      0207 1  
436      0208 1  
437      0209 1  
438      0200 1  
439      0201 1  
440      0202 1  
441      0203 1  
442      0204 1  
443      0205 1  
444      0206 1  
445      0207 1  
446      0208 1  
447      0209 1  
448      0200 1  
449      0201 1  
450      0202 1  
451      0203 1  
452      0204 1  
453      0205 1  
454      0206 1  
455      0207 1  
456      0208 1  
457      0209 1  
458      0200 1  
459      0201 1  
460      0202 1  
461      0203 1  
462      0204 1  
463      0205 1  
464      0206 1  
465      0207 1  
466      0208 1  
467      0209 1  
468      0200 1  
469      0201 1  
470      0202 1  
471      0203 1  
472      0204 1  
473      0205 1  
474      0206 1  
475      0207 1  
476      0208 1  
477      0209 1  
478      0200 1  
479      0201 1  
480      0202 1  
481      0203 1  
482      0204 1  
483      0205 1  
484      0206 1  
485      0207 1  
486      0208 1  
487      0209 1  
488      0200 1  
489      0201 1  
490      0202 1  
491      0203 1  
492      0204 1  
493      0205 1  
494      0206 1  
495      0207 1  
496      0208 1  
497      0209 1  
498      0200 1  
499      0201 1  
500      0202 1  
501      0203 1  
502      0204 1  
503      0205 1  
504      0206 1  
505      0207 1  
506      0208 1  
507      0209 1  
508      0200 1  
509      0201 1  
510      0202 1  
511      0203 1  
512      0204 1  
513      0205 1  
514      0206 1  
515      0207 1  
516      0208 1  
517      0209 1  
518      0200 1  
519      0201 1  
520      0202 1  
521      0203 1  
522      0204 1  
523      0205 1  
524      0206 1  
525      0207 1  
526      0208 1  
527      0209 1  
528      0200 1  
529      0201 1  
530      0202 1  
531      0203 1  
532      0204 1  
533      0205 1  
534      0206 1  
535      0207 1  
536      0208 1  
537      0209 1  
538      0200 1  
539      0201 1  
540      0202 1  
541      0203 1  
542      0204 1  
543      0205 1  
544      0206 1  
545      0207 1  
546      0208 1  
547      0209 1  
548      0200 1  
549      0201 1  
550      0202 1  
551      0203 1  
552      0204 1  
553      0205 1  
554      0206 1  
555      0207 1  
556      0208 1  
557      0209 1  
558      0200 1  
559      0201 1  
560      0202 1  
561      0203 1  
562      0204 1  
563      0205 1  
564      0206 1  
565      0207 1  
566      0208 1  
567      0209 1  
568      0200 1  
569      0201 1  
570      0202 1  
571      0203 1  
572      0204 1  
573      0205 1  
574      0206 1  
575      0207 1  
576      0208 1  
577      0209 1  
578      0200 1  
579      0201 1  
580      0202 1  
581      0203 1  
582      0204 1  
583      0205 1  
584      0206 1  
585      0207 1  
586      0208 1  
587      0209 1  
588      0200 1  
589      0201 1  
590      0202 1  
591      0203 1  
592      0204 1  
593      0205 1  
594      0206 1  
595      0207 1  
596      0208 1  
597      0209 1  
598      0200 1  
599      0201 1  
600      0202 1  
601      0203 1  
602      0204 1  
603      0205 1  
604      0206 1  
605      0207 1  
606      0208 1  
607      0209 1  
608      0200 1  
609      0201 1  
610      0202 1  
611      0203 1  
612      0204 1  
613      0205 1  
614      0206 1  
615      0207 1  
616      0208 1  
617      0209 1  
618      0200 1  
619      0201 1  
620      0202 1  
621      0203 1  
622      0204 1  
623      0205 1  
624      0206 1  
625      0207 1  
626      0208 1  
627      0209 1  
628      0200 1  
629      0201 1  
630      0202 1  
631      0203 1  
632      0204 1  
633      0205 1  
634      0206 1  
635      0207 1  
636      0208 1  
637      0209 1  
638      0200 1  
639      0201 1  
640      0202 1  
641      0203 1  
642      0204 1  
643      0205 1  
644      0206 1  
645      0207 1  
646      0208 1  
647      0209 1  
648      0200 1  
649      0201 1  
650      0202 1  
651      0203 1  
652      0204 1  
653      0205 1  
654      0206 1  
655      0207 1  
656      0208 1  
657      0209 1  
658      0200 1  
659      0201 1  
660      0202 1  
661      0203 1  
662      0204 1  
663      0205 1  
664      0206 1  
665      0207 1  
666      0208 1  
667      0209 1  
668      0200 1  
669      0201 1  
670     
```

```

158      0219  2      BEGIN
159      0220  2
160      0221  2
161      0222  2      LITERAL
162      0223  2      TRUE      = 1,
163      0224  2      DISP      = 0,
164      0225  2      DEC_IS_COMMAS_BIT = 1,
165      0226  2      BTZ_BIT    = 2,
166      0227  2      CONVERT_BIT = 32,
167      0228  2      COMMA_BIT   = 64,
168      0229  2      PROTECT_BIT = 256,
169      0230  2      RPG_BIT     = 2048;
170
171      0231  2
172      0232  2
173      0233  2
174      0234  2
175      0235  2
176      0236  2
177      0237  2
178      0238  2      LOCAL
179      0239  2      COB$SAB_PREV,
180      0240  2      DEV : WORD,
181
182      0241  2
183      0242  2
184      0243  2
185      0244  2
186      0245  2
187      0246  2
188      0247  2
189      0248  2      FLAG,
190      0249  2      NRO_FLAG : INITIAL (0),
191      0250  2      TEMP_NRO : BLOCK [12,BYTE] VOLATILE.
192      0251  2      TEMP_STRING : VECTOR [15,BYTE];
193      0252  2
194      0253  2
195      0254  2
196      0255  2
197      0256  2
198      0257  2      BUILTIN
199      0258  2      ACTUALCOUNT;
200      0259  2
201      0260  2      ++
202      0261  2      Get address of history of previous call.
203      0262  2      Note that using it this way will affect the whole longword
204      0263  2      at the address, and not just the first byte.
205      0264  2
206      0265  2
207      0266  2      -
208      0267  2      COB$SAB_PREV = COB$RET_A_AB_PREV();
209      0268  2
210      0269  2
211      0270  2      ++
212      0271  2      Set up FLAG parameter.
213      0272  3
214      0273  3      -
215      0274  4      FLAG = CONVERT_BIT + RPG_BIT;
216      0275  3      IF (.FLAGS AND DEC_IS_COMMAS_BIT) NEQ 0
217
218      0276  3      THEN FLAG = .FLAG + COMMA_BIT;
219
220      0277  3      ++
221      0278  3      Deal with DISPLAY_FIELD.
222      0279  3
223      0280  3      -
224      0281  3      IF .DISPLAY_FIELD NEQ 0
225      0282  3      THEN
226      0283  3      BEGIN
227      0284  3
228      0285  3      IF (.DISPLAY_FIELD[DSC$B_DTYPE] EQL DSC$K_DTYPE_NRO) AND
229      0286  3      ((.FLAGS AND BTZ_BIT) NEQ 0)
230      0287  3
231      0288  3      THEN
232
233      0289  3
234      0290  3
235      0291  3
236      0292  3
237      0293  3
238      0294  3
239      0295  3
240      0296  3
241      0297  3
242      0298  3
243      0299  3
244      0300  3
245      0301  3
246      0302  3
247      0303  3
248      0304  3
249      0305  3
250      0306  3
251      0307  3
252      0308  3
253      0309  3
254      0310  3
255      0311  3
256      0312  3
257      0313  3
258      0314  3
259      0315  3
260      0316  3
261      0317  3
262      0318  3
263      0319  3
264      0320  3
265      0321  3
266      0322  3
267      0323  3
268      0324  3
269      0325  3
270      0326  3
271      0327  3
272      0328  3
273      0329  3
274      0330  3
275      0331  3

```

```
215      0276 3
216      0277 3
217      0278 3
218      0279 3
219      0280 3
220      0281 3
221      0282 3
222      0283 3
223      0284 3
224      0285 3
225      0286 3
226      0287 3
227      0288 3
228      0289 3
229      0290 3
230      0291 3
231      0292 3
232      0293 3
233      0294 3
234      0295 3
235      0296 3
236      0297 3
237      0298 3
238      0299 3
239      0300 3
240      0301 4
241      0302 3
242      0303 3
243      0304 3
244      0305 3
245      0306 3
246      0307 3
247      0308 3
248      0309 3
249      0310 3
250      0311 3
251      0312 3
252      0313 3
253      0314 3
254      0315 3
255      0316 3
256      0317 3
257      0318 3
258      0319 3
259      0320 3
260      0321 4
261      0322 4
262      0323 4
263      0324 4
264      0325 4
265      0326 4
266      0327 4
267      0328 4
268      0329 4
269      0330 4
270      0331 4
271      0332 5

      + Convert blanks to zeroes if flag is set.
      CH$TRANSLATE (RPG$BTZ,
                      DISPLAY_FIELD[DSC$W_LENGTH], DISPLAY_FIELD[DSC$A_POINTER],
                      0, .DISP[AY_FIELD[DSC$W_LENGTH]], .DISP[AY_FIELD[DSC$A_POINTER]]);

      COB$SAB_PREV = DISP;           ! Set up history for proper advancing
      + Display the field.
      COB$DISP_SCR (1, .DISPLAY_FIELD, .FLAG);

      END;

      + Deal with ACCEPT_FIELD.
      IF (ACTUALCOUNT() EQL 3 AND .ACCEPT_FIELD NEQ 0)
      THEN
          BEGIN
              IF (.ACCEPT_FIELD[DSC$B_DTYPE] EQL DSC$K_DTYPE_NRO) AND
                  ((.FLAGS AND BTZ_BIT) NEQ 0)
              THEN
                  + Convert blanks to zeroes if flag is set.
                  CH$TRANSLATE (RPG$BTZ,
                                  ACCEPT_FIELD[DSC$W_LENGTH], ACCEPT_FIELD[DSC$A_POINTER],
                                  0, .ACCEPT_FIELD[DSC$W_LENGTH], .ACCEPT_FIELD[DSC$A_POINTER]);
                  COB$SAB_PREV = DISP;           ! Set up history for proper advancing
                  + Display the field.
                  COB$DISP_SCR (1, .ACCEPT_FIELD, .FLAG);

                  FLAG = .FLAG OR PROTECT_BIT;   ! Protect
                  WHILE TRUE DO
                      + Keep ACCEPTing until successful.
                      BEGIN
                          BIND
                          ZERO = UPLIT ('0');

                          COB$SAB_PREV = DISP;           ! Set up history for proper advancing
                          IF .ACCEPT_FIELD [DSC$B_DTYPE] EQL DSC$K_DTYPE_NRO
                          THEN
                              NRO_FLAG = TRUE;
                          IF .NRO_FLAG
                          THEN
                              BEGIN
```

```
272      0333 5
273      0334 5
274      0335 5
275      0336 5
276      0337 5
277      0338 5
278      0339 5
279      0340 5
280      0341 5
281      0342 5
282      0343 5
283      0344 6
284      0345 6
285      0346 6
286      0347 6
287      0348 6
288      0349 6
289      0350 6
290      0351 5
291      0352 5
292      0353 5
293      0354 5
294      0355 5
295      0356 5
296      0357 5
297      0358 5
298      0359 5
299      0360 5
300      0361 4
301      0362 4
302      0363 4
303      0364 4
304      0365 4
305      0366 6
306      0367 6
307      0368 5
308      0369 5
309      0370 4
310      0371 4
311      0372 5
312      0373 5
313      0374 5
314      0375 5
315      0376 5
316      0377 5
317      0378 5
318      0379 4
319      0380 4
320      0381 3
321      0382 3
322      0383 3
323      0384 3
324      0385 3
325      0386 3
326      0387 3
327      0388 3
328      0389 3

      + Overpunched numeric string must be saved because
      | COB$ACC SCR zeroes it out and if there is invalid
      | input, the original string must be available in
      | case the user responds to the reprompt with a <CR>.
      |
      | TEMP_NRO [DSCSW_LENGTH] = .ACCEPT_FIELD [DSCSW_LENGTH];
      | TEMP_NRO [DSCSB_DTYPE] = DSC$K_DTYPE_NRO;
      | TEMP_NRO [DSCSA_POINTER] = TEMP_STRING;
      | IF .ACCEPT_FIELD [DSC$B_CLASS] EQL DSC$K_CLASS_SD
      | THEN
      |   BEGIN
      |     TEMP_NRO [DSC$B_CLASS] = DSC$K_CLASS_SD;
      |     TEMP_NRO [DSC$B_SCALE] = .ACCEPT_FIELD [DSC$B_SCALE];
      |     TEMP_NRO [DSC$B_DIGITS] = .ACCEPT_FIELD [DSC$B_DIGITS];
      |
      |   END
      | ELSE
      |   TEMP_NRO [DSC$B_CLASS] = DSC$K_CLASS_S;
      |
      |   + Initialize the temporary string.
      |   |
      |   CH$MOVE (.ACCEPT_FIELD [DSCSW_LENGTH],
      |           .ACCEPT_FIELD [DSCSA_POINTER],
      |           TEMP_STRING);
      |
      | END;

      + Accept into the field.
      |
      | IF ( COB$ACC_SCR (0, (IF .NRO_FLAG
      |                      THEN TEMP_NRO
      |                      ELSE .ACCEPT_FIELD),
      |                      .FLAG, 0, 0, 0, 0) )
      |
      | THEN EXITLOOP
      | ELSE
      |   BEGIN
      |     + Error - print warning and re-prompt.
      |     |
      |     LIB$SIGNAL (RPG$_INVNUMENT);
      |     .COB$SAB_PREV = DISP;           ! Set up history for proper advancing
      |
      |   END;
      |
      | END;

      IF .NRO_FLAG
      THEN
      |
      | Move accepted string from temporary to actual accept field.
      |
      CH$MOVE (.ACCEPT_FIELD [DSCSW_LENGTH], TEMP_STRING,
               .ACCEPT_FIELD [DSC$A_POINTER]);
```

RPGSDSPLY
1-004

DSPLY an item
RPGSDSPLY - DSPLY an item

K 12
16-Sep-1984 02:13:33 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 13:04:17 [RPGRTL.SRC]RPGDSPLY.B32;1

Page 7
(4)

329 0390 3
330 0391 2
331 0392 2
332 0393 1

.TITLE RPG\$DSPLY DSPLY an item
.IDENT \1-004\

.PSECT _RPG\$CODE,NOWRT, SHR, PIC,2

00 00 00 30 00000 P.AAA: .ASCII \01<0><0><0>

ZERO=

P.AAA

.EXTRN COBSACC SCR, COB\$DISP_SCR
.EXTRN COB\$SRET_A_AB_PREV
.EXTRN LIB\$SIGNAL, RPGS_INVNUMENT
.EXTRN RPGSBTZ

.ENTRY RPGSDSPLY, Save R2,R3,R4,R5,R6,R7,R8,R9,-
R10,R11

0172

5B	00000000G	00	9E	00002	MOVAB	R10, R11
5A	00000000G	00	9E	00009	MOVAB	COB\$DISP_SCR, R11
5E		1C	C2	00010	SUBL2	RPGSBTZ, R10
		58	D4	00013	CLRL	#28, SP
00000000G	00	00	FB	00015	CALLS	NRO_FLAG
59		50	90	0001C	MOVL	#0, COBSSRET A AB_PREV
57	0820	8F	3C	0001F	MOVZWL	R0, COBSSAB_PREV
04	04	AC	E9	00024	BLBC	#2080, FLAG
57	40	A7	9E	00028	MOVAB	FLAGS, 1S
56	08	AC	D0	0002C	MOVL	64(R7), FLAG
		1D	13	00030	BEQL	DISPLAY_FIELD, R6
13	02	A6	91	00032	CMPB	3S
		0D	12	00036	BNEQ	2(R6), #19
04	6C	21	E1	00038	BBC	2S
04	B6	66	2E	0003C	MOVTC	#33, FLAGS, 2S
04	B6	66		00042		(R6), 24(R6), #0, RPGSB
		69	D4	00045	2\$: CLRL	(COBSSAB_PREV)
7E		56	7D	00047	MOVQ	R6, -(SPT)
		01	DD	0004A	PUSHL	#1
6B	03	FB	0004C		CALLS	#3, COB\$DISP_SCR
03	6C	91	0004F	3\$: CMPB	(AP), #3	
		01	13	00052	BEQL	4S
		04		00054	RET	
	0C	AC	D5	00055	4\$: TSTL	ACCEPT_FIELD
		01	12	00058	BNEQ	5S
		04		0005A	RET	
56	0C	AC	D0	0005B	5\$: MOVL	ACCEPT_FIELD, R6
13	02	A6	91	0005F	CMPB	2(R6), #19
		0D	12	00063	BNEQ	6S
04	6C	21	E1	00065	BBC	#33, FLAGS, 6S
04	B6	66	2E	00069	MOVTC	(R6), 24(R6), #0, RPGSB
04	B6	66		0006F		
		69	D4	00072	6\$: CLRL	(COBSSAB_PREV)
7E		56	7D	00074	MOVQ	R6, -(SPT)
		01	DD	00077	PUSHL	#1
6B	03	FB	00079		CALLS	#3, COB\$DISP_SCR

RPG\$DSPLY
1-004DSPLY an item
RPG\$DSPLY - DSPLY an item

L 12

16-Sep-1984 02:13:33
14-Sep-1984 13:04:17VAX-11 Bliss-32 V4.0-742
[RFGRTL.SRC]RPGDSPLY.B32;1Page 8
(4)

57	0100	8F	A8	0007C	7\$:	BISW2	#256, FLAG	: 0316	
13	02	69	D4	00081		CLRL	(COB\$SAB_PREV)	: 0326	
		13	A6	91	00083	CMPB	2(R6), #T9	: 0327	
				03	12	BNEQ	8S	: 0329	
				58	01	MOVL	#1, NRO FLAG	: 0330	
10	AE	2B	58	E9	0008C	BLBC	NR0 FLAG, 11\$: 0339	
12	AE	66	B0	0008F		MOVW	(R6), TEMP_NRO	: 0340	
14	AE	13	90	00093		MOVB	#19, TEMP_NRO+2	: 0341	
	09	09	6E	9E	00097	MOVAB	TEMP_STRING, TEMP_NRO+4	: 0342	
				A6	91	CMPB	3(R6), #9	: 0346	
				10	12	BNEQ	9S	: 0347	
13	AE	08	09	90	000A1	MOVB	#9, TEMP_NRO+3	: 0348	
18	AE	09	A6	90	000A5	MOVB	8(R6), TEMP_NRO+8	: 0349	
19	AE		A6	90	000AA	MOVB	9(R6), TEMP_NRO+9	: 0350	
				04	11	BRB	10\$: 0352	
6E	13	AE	04	B6	01	90	000B1	9\$:	: 0357
				66	28	000B5	10\$:	: 0366	
				7E	7C	000BA	11\$:	: 0369	
				7E	7C	000BC	CLRQ	-(SP)	
				57	DD	000BE	CLRQ	-(SP)	
			08	58	E9	000C0	PUSHL	FLAG	: 0368
			50	AE	9E	000C3	BLBC	NRO FLAG, 12\$: 0369
				50	DD	000C7	MOVAB	TEMP_NRO, R0	: 0370
				02	11	000C9	PUSHL	R0	: 0371
				56	DD	000CB	12\$:	13\$: 0372
				7E	D4	000CD	13\$:	R6	: 0373
				07	FB	000CF	CLRL	-(SP)	: 0374
00000000G	00			50	E8	000D6	CALLS	#7, COBSACC_SCR	: 0375
	11			8F	DD	000D9	BLBS	R0, 14\$: 0376
00000000G	00	00000000G		01	FB	000DF	PUSHL	#RPG\$ INVNUMENT	: 0377
				69	D4	000E6	CALLS	#1, LTBS\$SIGNAL	: 0378
				97	11	000E8	CLRL	(COB\$SAB_PREV)	: 0379
			05	58	E9	000EA	14\$:	7S	: 0380
04	B6		6E	66	28	000ED	BLBC	NRO FLAG, 15\$: 0381
				04	000F2	15\$:	MOV3	(R6), TEMP_STRING, 24(R6)	: 0382
							RET		: 0383
									: 0384
									: 0385
									: 0386

: Routine Size: 243 bytes. Routine Base: _RPG\$CODE + 0004

: 333 0394 1
: 334 0395 0 END ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes
_RPG\$CODE	247	NOVEC,NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

RPG\$DSPLY
1-004

DSPLY an item
RPG\$DSPLY - DSPLY an item

M 12
16-Sep-1984 02:13:33
14-Sep-1984 13:04:17

VAX-11 Bliss-32 V4.0-742
[RPGRTL.SRC]RPGDSPLY.B32;1

Page 9
(4)

File	----- Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
-\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	9	0	581	00:01.0
-\$255\$DUA28:[RPGRTL.OBJ]RPGLIB.L32;1	54	0	0	9	00:00.1

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LISS:RPGDSPLY/OBJ=OBJ\$:RPGDSPLY MSRC\$:RPGDSPLY/UPDATE=(ENH\$:RPGDSPLY)

Size: 243 code + 4 data bytes
Run Time: 00:07.3
Elapsed Time: 00:25.5
Lines/CPU Min: 3259
Lexemes/CPU-Min: 14079
Memory Used: 103 pages
Compilation Complete

0331 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

